**需求设计初稿**

**数据结构概要设计：**

1. 首先有用户(user)，每个用户可以创建多个数据分析作业(job)，在每个作业(job)中，有同一组数据和不同算法组合在一起产生的活动(activity)，对应的训练数据模型(TrainDataFrame)，预测数据模型(ProDataFrame)，算法模型(model)，算法(algo)

**数据结构详细设计**

//用户

User{

Int id;//主键，对用户透明，

String account;//用户账户

String password//密码

}

//作业

Job{

Int jobID;//应用id，这个也可以作为主键

String appName;//应用名称，用户自己起的

Int userid；//这个job属于那个用户

String srcDatafile;//源数据文件路径

String dataFrame;//格式化文件路径

List<activity> activities;//用户在这个app中创建的活动的集合

}

//活动

Activity{

Int ActivityID；//活动主键，

String activityName;//作业名称，我们生成的，用来调度集群的时候使用,使用所属的应用名称加活动编号生成。

String jobID;//属于哪个作业

Int num;//这个活动就不需要名称了，有个编号就可以了，在某个应用中对应几号活动。

String algoName;//算法名称

String state;//这个活动的状态，判断是否训练预测结束，由于训练过程和主程序是兵法云行的所以需要多进程编程。

Cluster Info;//这里由于可能需要使用到运行过程中这个工作的相关参数，所以这个用来返回集群的信息。这个不应该添加到数据结构里，应该在接口中直接获取，因为怕忘了，先写在这里。

}

//数据模型

//因为要根据这个数据结构生成数据库

//这个生成过程最好要使用MR，在hadoop集群上运行，不要在电脑上运行，数据特别多的时候，算不了，MR编程目前还不会，所以后面需要学一下

//最初的想法是把用户的数据以文件的形式保存在hadoop上，然后使用MR编程，根据用户的输入，将用户在hadoop上的文件，在处理成下面dataframe这样的文本文件，这样的话其实是不需要下面这个数据结构，因为不需要存库

//如果需要存库的话，也和上面这个过程差不多，这是用来操作大数据的数据库，我目前没有学，使用mysql不知道会不会影响效率。而且不知道hadoop能不能操作mysql,好像是能。

//没有设计数据存储的地方，所以到时候，数据的预览如果使用文件的话，需要用到数据的时候就直接使用二位链表取出来即可以了。

DataFrame{

List<String> labels;//数据标签

List<String> features;//数据特征

}

Model{

String modelName;//自己对应的模型的名称

String jobID;//自己对应的活动的编号

String algoName;//对应的算法的名称

String path;//模型的存储路径

}

//算法：这个好像就只记录个名字和jar文件存储目录可以了

Algo{

String algoName;

String path;

}

**数据库设计：**

针对上面进行的数据库需求设计设计数据库表

**系统结构：**

首先有系统预先编写好相关算法的jar包上传到集群当中的特定算法存储空间，然后用户上传csv数据文件后，目前有两种方案

1. 系统将csv文件上传到hadoop上，然后使用MR编程，对csv文件进行数据处理，有用户选择将文件中的那些字段作为label那些字段作为feature，然后使用MR编程通过获取到的参数将csv文件调整为<label，feature>的形式，这个已经找到解决方案，不用担心，在sparkMlib书上，有解决方案
2. 系统将csv文件上传到hadoop，使用MR编程将数据以<label,feature>的格式存入数据库，然后再scala程序中从数据库中读取参数进行训练。
3. 训练结束后产生的模型，将保存到集群中hadoop上的一个目录中，然后并且将文件模型路径保存到数据库中。